

Embedded Open Source Summit 2024

# Rethinking U-Boot Devicetree Story

Sumit Garg | Linaro Ltd.

17-04-2024



# Who am I?

- Senior systems developer at Linaro
  - 9 years of experience
- Keen interest in platform security
- Contributions across the embedded software stack be it firmwares, bootloaders, trusted OS, Linux kernel, OpenEmbedded and so on.
- Have worked on platforms from various silicon vendors: Qualcomm, NXP, TI, Renesas, Socionext and so on.

# Linaro is the software engine of the Arm Ecosystem

Linaro empowers rapid product deployment within the dynamic Arm Ecosystem.

- Our cutting-edge solutions, services and collaborative platforms facilitate the swift development, testing, and delivery of Arm-based innovations, enabling businesses to stay ahead in today's competitive technology landscape.
- Our expertise and contributions spread from Testing & LTS, Security, Cloud & Edge Computing, IoT, AI, CI/CD, Toolchain and Virtualization to vertical projects like Windows on Arm and Android Ecosystem enabling and maintenance.
- Linaro fosters an environment of collaboration, standardization and optimization among businesses and open source ecosystems to accelerate the deployment of Arm-based products and technologies along with representing a pivotal role in open source discovery and adoption.

**Linaro has enabled trust, quality and collaboration since 2010**

# How to describe devicetree?

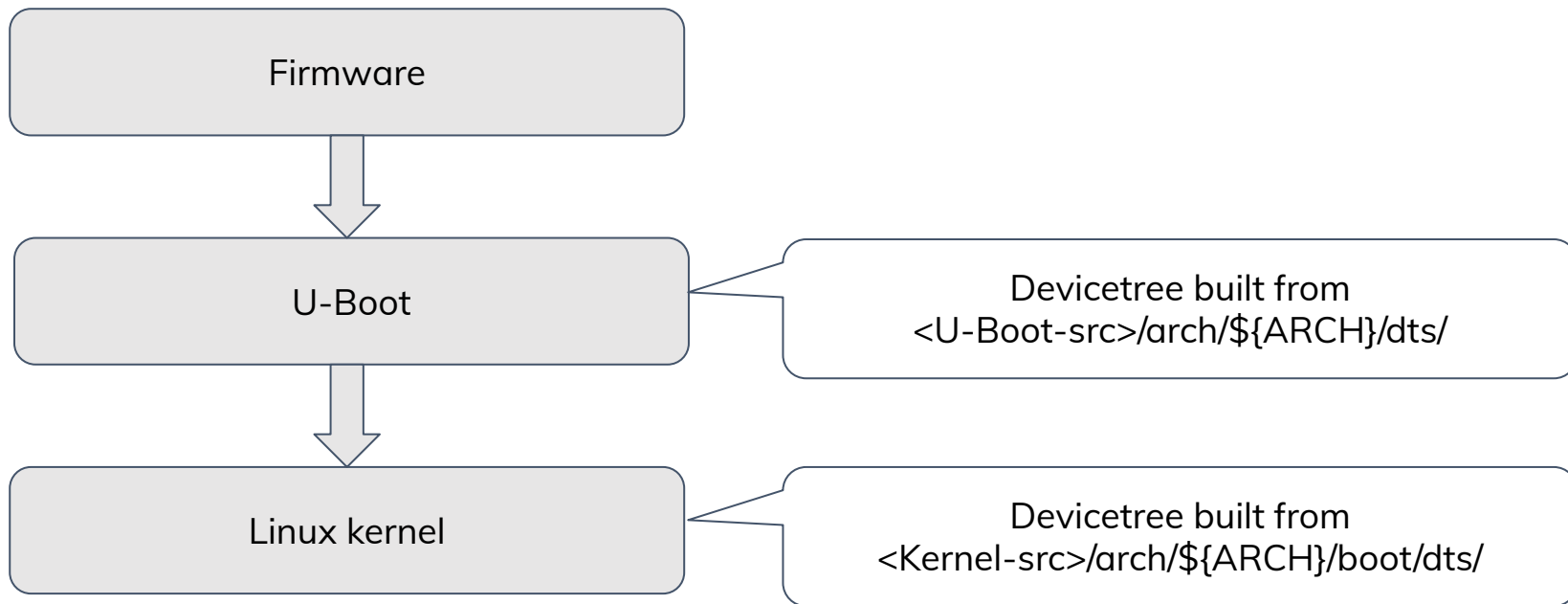
A devicetree is a **tree data structure with nodes** that describe the devices in a system.

The **Devicetree Bindings** specifies requirements as to how specific types and classes of devices are represented in the devicetree.

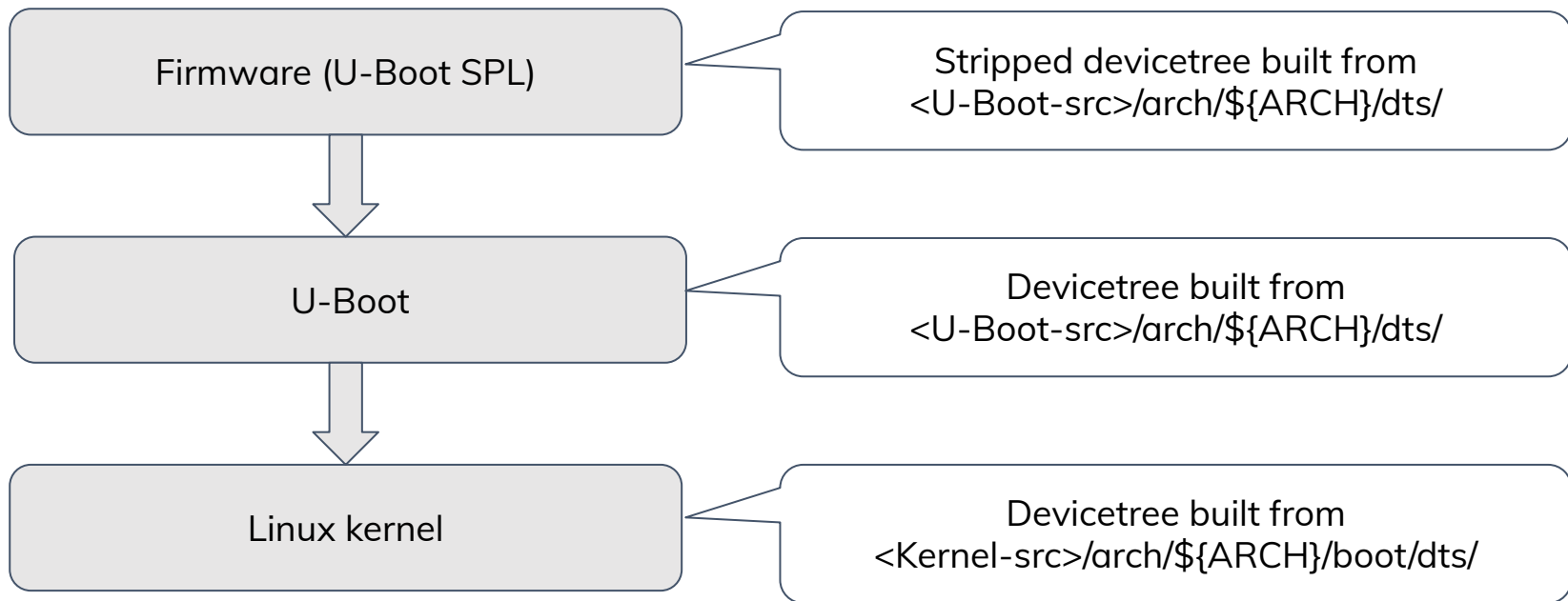
A DTSpec-compliant devicetree describes device information in a system that **cannot necessarily be dynamically detected** by a client program.

Further reading: <https://devicetree-specification.readthedocs.io/en/latest/index.html>

# Where does the devicetree come from?

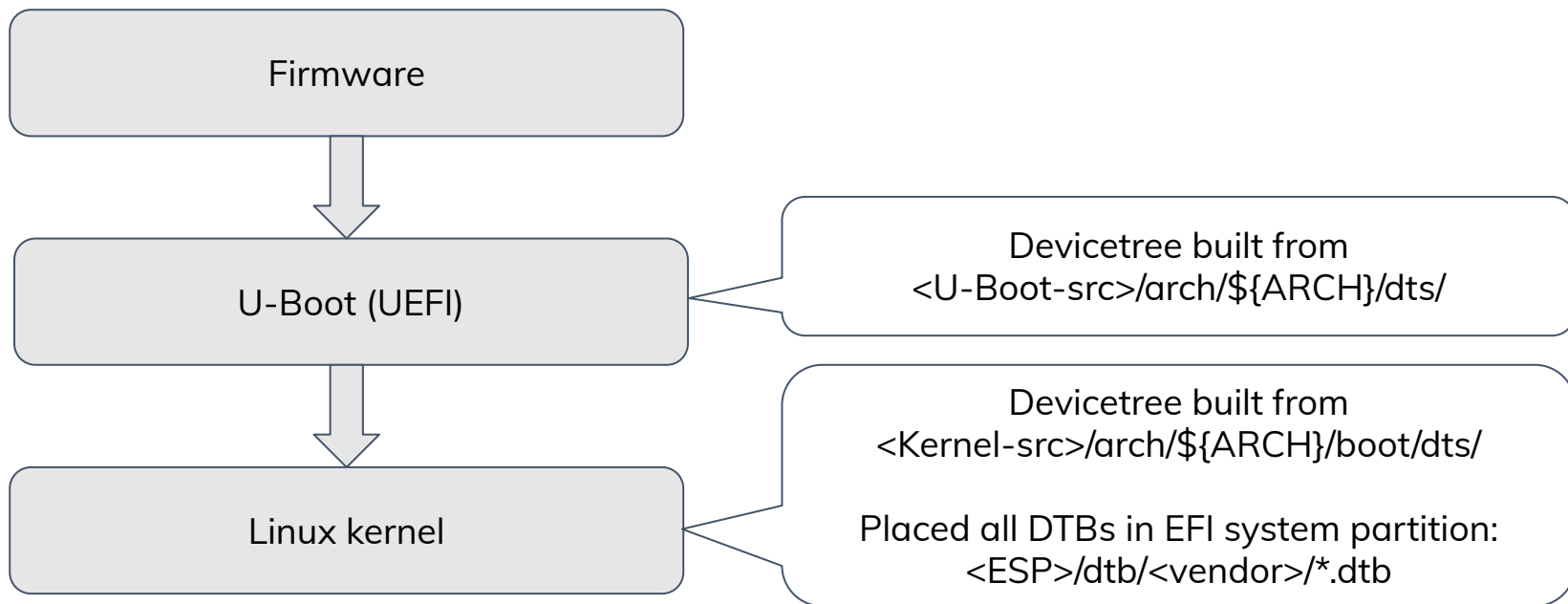


# Where does the devicetree come from? - II



# Where does the devicetree come from? - |||

Standard Linux distributions



# Devicetree fragmentation

## Linux 6.7 (Jan 7th, 2024)

- De-facto source for devicetree source (DTS) files
  - `arch/${ARCH}/boot/dts`
- De-facto source for device specific bindings
  - `Documentation/devicetree/bindings`

## U-Boot v2024.01 (Jan 8th, 2024)

- Manual DTS files import from Linux kernel
  - `*-u-boot.dtsi` files to customize import
  - **Scattered** import history per vendor/SoC: Linux kernel v6.7-rc1, v6.7-rockchip-dts64-1, v6.6-rc5, v6.6-rc6 and so on...
- Manual device specific bindings import from Linux kernel
  - Very **rarely** synced
  - No bindings check



# Devicetree fragmentation - ||

## Linux 6.7 (Jan 7th, 2024)

```
./include/dt-bindings/clock/rk3399-cru.h
```

```
...
```

```
#define SCLK_PCIEPHY_REF100M      167
```

```
#define SCLK_DDRC                  168
```

```
#define SCLK_TESTCLKOUT1          169
```

```
#define SCLK_TESTCLKOUT2          170
```

```
...
```

## U-Boot v2024.01 (Jan 8th, 2024)

```
./include/dt-bindings/clock/rk3399-cru.h
```

```
...
```

```
#define SCLK_PCIEPHY_REF100M      167
```

```
#define SCLK_USBPHY0_480M_SRC      168
```

```
#define SCLK_USBPHY1_480M_SRC      169
```

```
#define SCLK_DDRCLK                170
```

```
#define SCLK_TESTOUT2              171
```

```
...
```

# Devicetree fragmentation - |||

**Linux 6.7 (Jan 7th, 2024)**

**U-Boot v2024.01 (Jan 8th, 2024)**

De-facto standard Qcom device bindings:

Custom Qcom device bindings:

.../bindings/serial/qcom,msm-uartdm.yaml

.../device-tree-bindings/serial/msm-serial.txt

.../bindings/pinctrl/qcom,msm8916-pinctrl.yaml

.../device-tree-bindings/gpio/gpio-msm.txt

.../bindings/pinctrl/qcom,msm8996-pinctrl.yaml

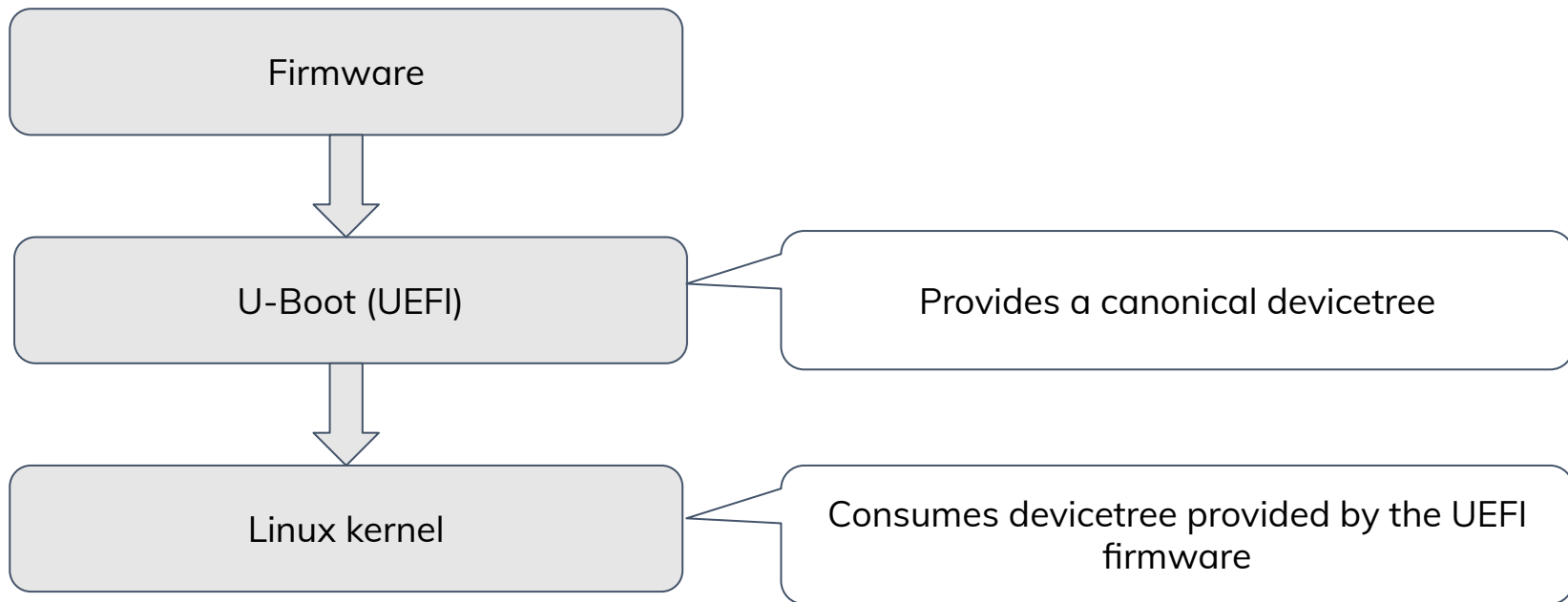
.../device-tree-bindings/mmc/msm\_sdhci.txt

.../bindings/mmc/sdhci-msm.yaml

# U-Boot maintainer's nightmare

- Are huge **manual DTS files syncs** easily reviewable?
  - Especially if the Linux kernel source version differs among syncs.
- How U-Boot maintainers should **enforce DT bindings checks**?
- How should we **bring uniformity** in the DTS files synced from the Linux kernel?
- Can we really boot Linux with the **same DTB** as used by U-Boot?
- As part of Secure boot, do I need to **authenticate multiple DTBs** for the same hardware?

# What do standards (EBBR, SystemReady IR) say?



# Benefits of a canonical devicetree

- A **single devicetree source** file describing system hardware
  - Allows to support the common devicetree source file across various versions of consumer projects
- Linux standard distributions **don't have to worry about devicetree packaging**
- Devicetree blob can be **authenticated as part of firmware**
  - No custom authentication required for devicetree blob
- **Reduced devicetree maintenance** effort for consumer projects
  - Currently Linux kernel hosts devicetree sources
- **Avoids** consumer projects to add project specific **runtime/configuration data** to devicetree

# Where my journey began?

- Tasked to add U-Boot to the boot chain on **Qualcomm platforms**
  - Proprietary bootloaders are hard to deal with especially the signed ones
- Had to deal with **heavily customized** devicetree source (DTS) files
  - U-Boot specific custom bindings
- Step by step conversion of U-Boot drivers to comply with upstream devicetree bindings
- Until my colleague Caleb took on the [challenge](#) of directly importing DTS files from Linux kernel and customizing them for U-Boot needs

# Where my journey began? - II

- Got involved into **discussions with Linux DT bindings maintainers** regarding how we can improve U-Boot DT situation.
- [Rob Herring](#) turned my attention to [devicetree-rebasing](#) repository which can help improve U-Boot DT usage.

# Where my journey began? - II

- Got involved into **discussions with Linux DT bindings maintainers** regarding how we can improve U-Boot DT situation.
- [Rob Herring](#) turned my attention to [devicetree-rebasing](#) repository which can help improve U-Boot DT usage.

The devicetree-rebasing contains **devicetree bindings and sources extracted** from the Linux kernel source tree. The repo is updated at every Linux kernel major release or intermediate release candidates.



# A step towards canonical devicetree

I started with an effort to **automatically import** DT bindings as well as DTS files in U-Boot from devicetree-rebasing repository.

Basically meant:

- Adding devicetree-rebasing repository as a **git subtree within U-Boot repository** (**dts/upstream** sub-directory).
- Augmenting DTS files with U-Boot specific stuff via **\*-u-boot.dtsi**.
- Added a Kconfig option: **OF\_UPSTREAM**
  - To make it an opt-in choice for SoC maintainers.
  - Maintain backwards compatibility.
- Added bindings check in U-Boot
  - **dtbs\_check** as a Makefile rule.

# dtbs\_check in U-Boot

With **dtb/upstream** git subtree, the devicetree bindings are also regularly synced with Linux kernel as **dtb/upstream/Bindings**. Allows U-Boot to run devicetree bindings schema checks to bring compliance among U-Boot core/drivers regarding usage of devicetree.

Perform validation of **complete DTB files**:

```
$ make dtbs_check
```

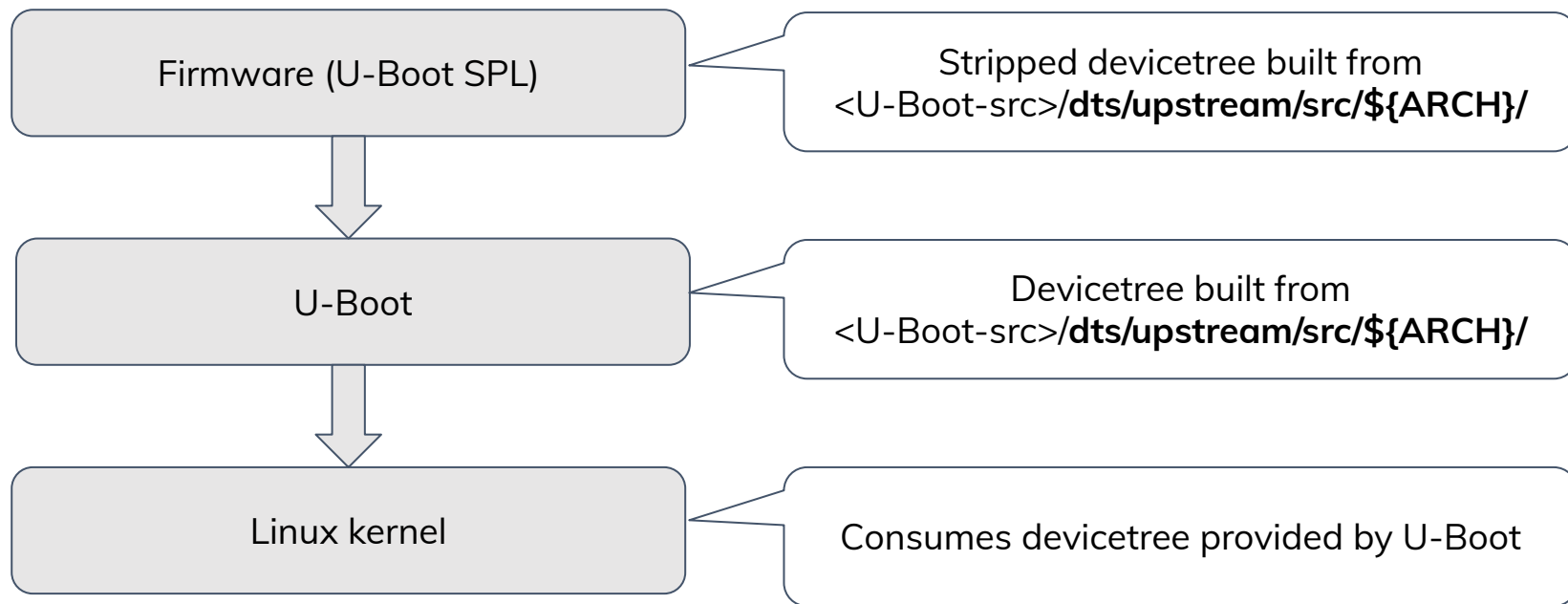
Perform validation with a **subset of matching schema files**:

```
$ make dtbs_check DT_SCHEMA_FILES=trivial-devices.yaml:rtc.yaml
```

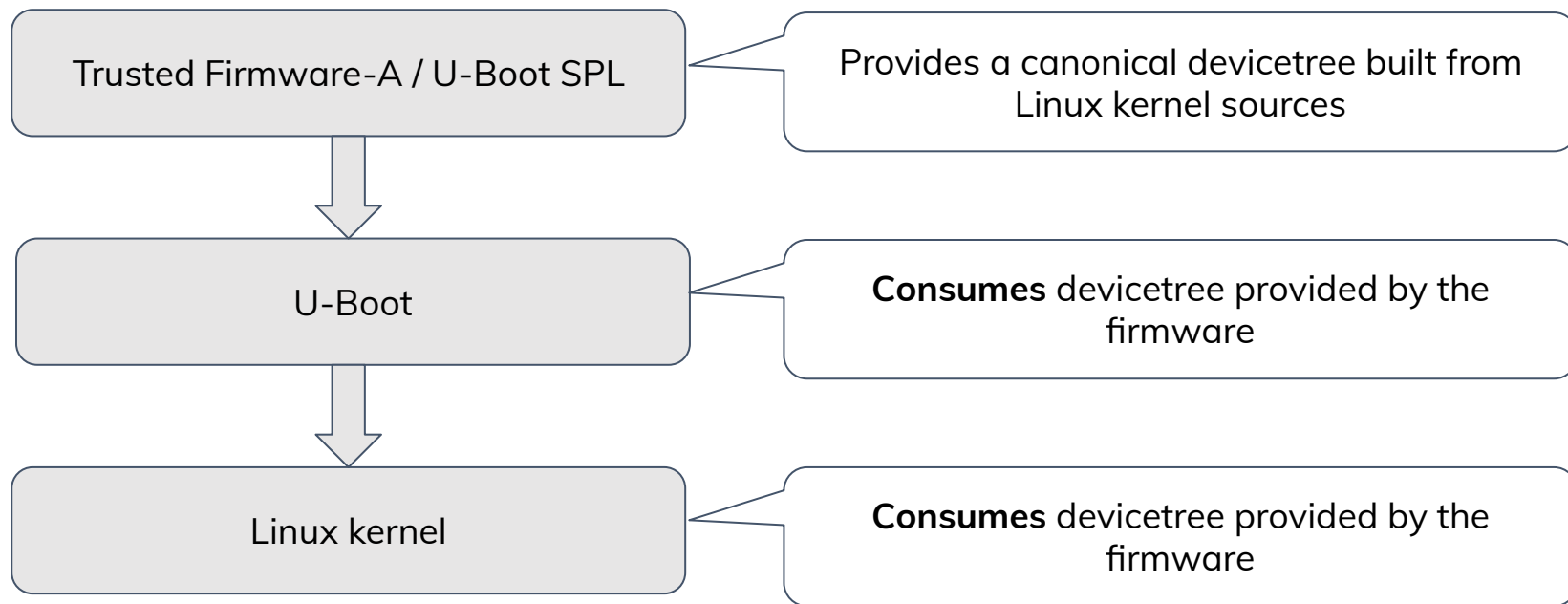
```
$ make dtbs_check DT_SCHEMA_FILES=/gpio/
```

```
$ make dtbs_check DT_SCHEMA_FILES=trivial-devices.yaml
```

# The future for devicetree (OF\_UPSTREAM)



# The future for devicetree on Arm



# Challenges with OF\_UPSTREAM

Do we really have a **stable devicetree ABI**?

<https://docs.kernel.org/devicetree/bindings/ABI.html>

Possible reasons for DT ABI breakage:

- The **DT incorrectly described the hardware**, “sorry not sorry” it's an incompatible DTS change now.
- **Early stages of DT development** for a particular SoC with a limited user-base.
- Bringing **DT bindings compatibility** for a legacy DT.
- Just for **improving DT bindings look and feel**, argument being nobody else apart from Linux kernel cares.

# Mitigations to avoid DT ABI breakages

- **dts/upstream** subtree synced only at major (.0) Linux kernel releases.
- **dts/upstream** subtree sync only happens after the U-Boot next branch opens
  - Provides ample time for developers to fix any problems observed.
- [Rob Herring](#) is already working on a [DT ABI checking tool](#). Feedback is very much welcome regarding what could be considered as a DT ABI breakage.
- Proposal for a **Linux kernel sub-arch maintainers profile**
  - To keep them officially aware that U-Boot aggressively syncs DT sources from Linux kernel.

# Challenges with OF\_UPSTREAM - II

**Challenge:** How do we **handle fixes to DTs** which are applied to Linux kernel stable releases?

**Mitigation:** dts/upstream subtree allows to cherry pick DT fixes if they are required for a particular U-Boot platform.

# Challenges with OF\_UPSTREAM - II

**Challenge:** How do we **handle fixes to DTs** which are applied to Linux kernel stable releases?

**Mitigation:** dts/upstream subtree allows to cherry pick DT fixes if they are required for a particular U-Boot platform.

**Challenge:** How to handle **updates for \*-u-boot.dtsi** after automatic DT syncs?

**Mitigation:** Try to minimize **\*-u-boot.dtsi** files or better try to get rid of them.



# Challenges with OF\_UPSTREAM - II

**Challenge:** How do we **handle fixes to DTs** which are applied to Linux kernel stable releases?

**Mitigation:** dts/upstream subtree allows to cherry pick DT fixes if they are required for a particular U-Boot platform.

**Challenge:** How to handle **updates for \*-u-boot.dtsi** after automatic DT syncs?

**Mitigation:** Try to minimize **\*-u-boot.dtsi** files or better try to get rid of them.

Many of the challenges were brought up during the review process, so **thanks to the reviewers** for bringing them up.

# Next steps...

- Let's work collaboratively towards a **coherent devicetree ecosystem** for the embedded community
  - A special callout to Linux kernel and U-Boot sub-arch maintainers.
- Let's switch to **OF\_UPSTREAM** in U-Boot, platforms already migrated or are in the process of migration:
  - Amlogic
  - Qualcomm
  - Marvell
  - NXP
  - Renesas
  - Rockchip
  - TI
  - ...

# Thank you

Visit [www.linaro.org](http://www.linaro.org)

Reach out to me at [sumit.garg@linaro.org](mailto:sumit.garg@linaro.org)

